Five-axis contour error control based on spatial iterative learning

Article (Accepted Version)

Five-axis contour error control based on spatial iterative learning

Jiangang Li, Senior Member, IEEE, Zhiyang You, Yanan Li*, Senior Member, IEEE, Enming Miao and Ruijie Yue

Abstract—In this paper, a contour error control strategy based on spatial iterative learning control (sILC) is developed for repetitive processing of five-axis computer numerical control (CNC) machine tools. A curve approximation method with an adaptive moving window is developed to achieve accurate tool position and orientation contour error estimation, and a five-axis contour error control strategy based on sILC is proposed. The compensation method is derived using an sILC algorithm to modify the geometric reference path instead of modifying the controller. The experimental results show that the proposed control strategy reduces contour errors of five-axis CNC machine tools, and it outperforms the traditional tracking error control.

Note to Practitioners—This paper aims to propose an effective five-axis contour error control scheme. At present, most of the five-axis contour error control methods rely on the modification of the controller, which is not allowed in commercial CNC systems. Therefore, we propose a five-axis contour error compensation algorithm based on sILC, which modifies the system's reference path. Experimental results show that the proposed method can reduce the five-axis contour errors, while maintaining the machining efficiency.

Index Terms—Five-axis contour error estimation, spatial iterative learning control, five-axis contour error control

I. Introduction

Five-axis computer numerical control (CNC) machine tools are widely used in the processing of complex curved components in industries such as aerospace and marine. Achieving high-speed, high-precision contour control is one of the important goals of these applications. Due to fact such as mismatches of multiple axes, a contour error between the desired path and actual path is found to be inevitable at the tool tip. In recent decades, researchers have done lots of research in order to reduce the contour error [1], [2], [3].

One method to reduce the contour error is through reducing the single-axis tracking error, i.e. tracking control. Designing a controller to improve tracking accuracy is a common control problem in many fields [4], [5], [6], [7]. A sliding mode controller was introduced in [8], which reduced the tracking error of each axis when external disturbances such as friction and cutting force exist. For dismissing the adverse influences of cutting, friction force and parameter uncertainties, a closed-loop pole configuration controller with disturbance elimination was proposed in [9]. [10] presented a tracking error pre-compensation method based on feedforward friction compensation, which effectively improved the tracking accuracy of a single feed axis. However, in multi-axis machining, reducing the tracking error does not necessarily reduce the contour error [11], [12], and in some cases, it may even increase the contour error [1]. Therefore, the contour error control method is investigated to directly reduce the contour error during the machining process. [13] estimated the contour error using the linear relationship between the tracking errors, and designed a variable gain cross-coupling controller, thereby improving the estimation accuracy of some nonlinear trajectory contour error. [14] used the second-order Taylor series expansion to estimate the contour error as the shortest distance from the actual position to the desired path, and designed a controller based on closed-loop position control and cross-coupling control. [15] designed a contour error controller based on speed compensation with the idea of cross-coupling control.

Iterative learning control (ILC) has been successfully applied in many fields [16], [17], [18], [19]. It has been used for repetitive processing of CNC machine tools, which enables the controllers to learn information from previous iterations and update control inputs until the required accuracy is achieved. [20] proposed a contour error control method that combines cross-coupled ILC and single-axis ILC. [21] proposed command-based iterative learning control (cILC), which updated input commands without changing the original control structure. [22] proposed a contour error control method based on spatial iterative learning control (sILC), which transformed the tracking error in the time domain to the contour error in the spatial domain.

Five-axis machine tool is currently widely used in industry because of its flexibility [23], [24], where five-axis contour control is developed to reduce the machining error. [25] designed a double sliding mode controller to effectively reduce the five-axis contour errors. [26] considered the geometric characteristics of the five-axis

This work was supported in part by the National Key R&D Program of China under Grant 2019YFB1703700 and in part by Shenzhen Science and Technology Program under Grant GXWD20201230155427003-20200821171505003.

J. Li, Z. You and R. Yue are with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, Shenzhen, China 518055.

*Correspondence: Y. Li is with the Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, UK Email: yl557@sussex.ac.uk

E. Miao is with the Department of Mechanical Engineering at Chongqing University of Technology, Chongqing, China 400054.

tool path with smooth corners to compensate for the contour errors in real time. [27] combined the proposed contour error estimation algorithm based on the three-point arc approximation with the contour error controller. [28] proposed to shape the trajectory to suppress residual vibration so as to compensate for the contour errors. [29] used the idea of model predictive control (MPC) to adjust the position point, retaining the axis speed and acceleration constraints. [30] proposed a command-based ILC for contour error control of five-axis machine tools. [31] proposed an off-line gain adjustment (OGA) method to reduce the contour error in five-axis machining.

In this article, five-axis contour error control based on spatial iterative learning is proposed to improve the repeated machining accuracy of five-axis NC systems. Compared with the above works, this study proposes a contour error estimation method based on the actual five-axis contour error definition [32] and a new five-axis contour error control strategy based on sILC. The reason for choosing the definition of actual contour error is that the iterative control is carried out in an off-line manner, and the contour error compensation depends on the modification of the reference position based on the desired position, rather than the actual position. Compared with the control method of modifying the controller parameters, the method of modifying the path can preserve the stability of the system and is applicable to most of the off-the-shelf systems. Compared with [28] that also modifies the reference path, we supplement the control of the tool orientation contour error. Different from [30] that also used ILC to control contour error, our sILC considers spatial nature of the contour error, and its convergence and stability will be proved. Finally, we compare the tracking error control with our sILC, whose superiority will be demonstrated by experimental results. We summarize three contributions of this work as follows:

1) A curve approximation method with an adaptive moving window is presented to achieve accurate tool position and orientation contour error estimation.

2) A five-axis contour error control strategy based on sILC is proposed to effectively control contour errors of five axes.

3) The comparative experiments using the tracking control and the contour control proposed in this article are implemented to verify the effectiveness of the proposed method.

In the second section, we improve a five-axis contour error estimation algorithm; in the third section, we describe the geometric reference path modification method based on sILC, and theoretically prove its convergence and stability; in the fourth section, we design comparative experiments using a five-axis machine tool to verify the effectiveness of the proposed method; finally, the fifth section summarizes this work.

II. Five-axis Contour Error Estimation

In five-axis CNC systems, the contour error includes the tool tip position error and tool orientation error. Both of these two errors will have a significant effect on the machining accuracy of the workpiece, so it is necessary to accurately estimate them.

For this purpose, an improved definition of five-axis contour errors is proposed in this paper based on [32], as illustrated in Fig. 1. (P_d, O_d) are respectively the tool tip position and orientation coordinates on the desired path; (P_a, O_a) are respectively the tool tip position and orientation coordinates on the actual path; P_c is the point closest to the desired position P_d; O_c is the tool orientation corresponding to the point P_c on the actual path; e_p is the tracking error; ε_p is the tool tip position contour error; and ε_o is the tool orientation contour error (the tool orientation contour error is regarded as the angle deviation between the tool axis vector O_c and the tool axis vector O_d).

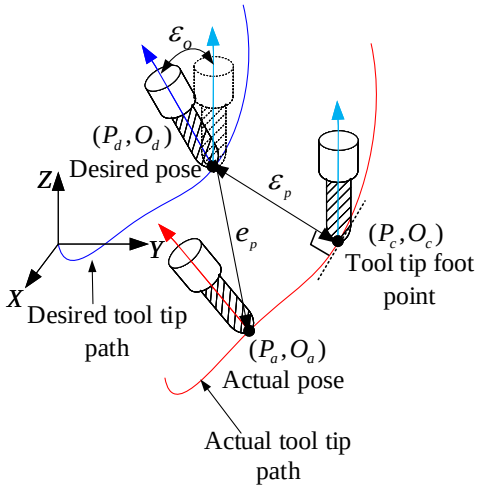


Fig. 1: Schematic diagram of five-axis contour error definition

A. Tool Tip Position Contour Error Estimation

For the estimation of the tool tip position contour error, we search the closest point and calculate the tool tip foot point of the contour error vector. The specific estimation process is introduced as follows.

For any desired trajectory in space, the actual positions can be obtained after running of the CNC machine tool. As shown in Fig. 2, $P_{d+i}, i \in [-3, 3]$ forms the desired path and $P_{a+i}, i \in [-3, 3]$ forms the actual path. First, we search the actual position point P_a closest to the desired position point P_d, and regard it as a temporary search point.

Then, we use a moving window to determine the search range around the temporary search point. If the moving window size is fixed, the above search method may not be applicable to the contour error estimation at the large curvature area on the path. As illustrated in Fig. 3, although $\left|\overrightarrow{P_d P_f}\right| < \left|\overrightarrow{P_d P_a}\right|$ where P_f, P_a are two points on the actual path, the contour error should be estimated as $\left|\overrightarrow{P_d P_a}\right|$. To address this issue, a numerical algorithm based on an adaptive moving window is designed to search two

actual position points closest to the desired one. The time window size is set as $n = \frac{\left|\overrightarrow{P_d P_a}\right|}{\left|\overrightarrow{P_a P_{a-1}}\right|}$. Then, according to the size of the moving window, the actual position point $P_{a+j} \in [P_{a-n}, P_{a+n}]$ closest to the desired one P_d is found.

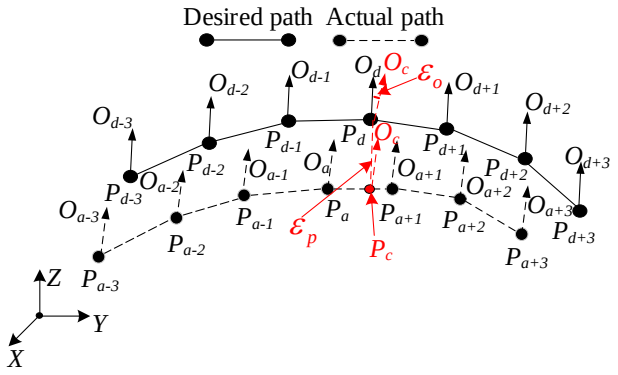


Fig. 2: Desired path and actual path for a five-axis machine tool

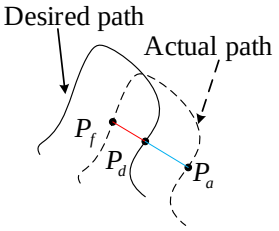


Fig. 3: Illustration of wrong search at a large curvature area of the path

With P_{a+j}, tool position contour error foot point P_c can be calculated. By connecting P_{a+j} with P_{a+j+1} and P_{a+j-1} respectively, two straight lines with lengths $L_1 = \left|\overrightarrow{P_{a+j-1}P_{a+j}}\right|$, $L_2 = \left|\overrightarrow{P_{a+j}P_{a+j+1}}\right|$ are obtained. Fig. 4 shows the line with L_1, while the line with L_2 is similar and thus omitted.

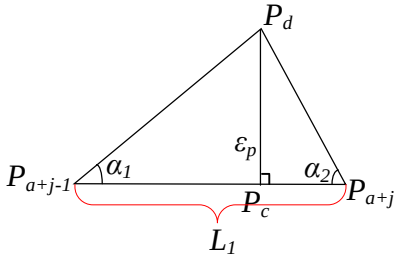


Fig. 4: Calculation of tool position contour error foot point

With $\left|\overrightarrow{P_{a+j-1}P_d}\right|$, $\left|\overrightarrow{P_d P_{a+j}}\right|$ and $\left|\overrightarrow{P_{a+j-1}P_{a+j}}\right|$ available, the values of angles α_1 and α_2 can be calculated by the vector angle formula. And then the coordinates of the foot point P_c can be obtained. To determine the foot point P_c of the line, we need to discuss the situations as follows:

Case 1: If $\alpha_1 < 90^o$ or $\alpha_2 < 90^o$, there is a foot point P_c on the line so that

$$\begin{cases} \left|\overrightarrow{P_d P_c}\right| = \left|\overrightarrow{P_d P_{a+j-1}}\right| \sin\alpha_1 \\ \left|\overrightarrow{P_{a+j-1}P_c}\right| = \left|\overrightarrow{P_d P_{a+j-1}}\right| \cos\alpha_1 \end{cases} \tag{1}$$

Then, the coordinate of the foot point P_c can be calculated as

$$P_c = P_{a+j-1} + \frac{\left|\overrightarrow{P_{a+j-1}P_c}\right|}{\left|\overrightarrow{P_{a+j-1}P_{a+j}}\right|} \overrightarrow{P_{a+j-1}P_{a+j}} \tag{2}$$

Case 2: If $\alpha_1 = 90^o$ or $\alpha_2 = 90^o$, then the foot point is P_{a+j-1} or P_{a+j}.

Case 3: If $\alpha_1 > 90^o$ and $\alpha_2 > 90^o$, then there is no foot point on the line.

If there are two foot points on both the lines with L_1 and L_2, then the foot point with a shorter distance to P_d is selected; if there is a foot point on only one line, then P_c is the foot point on this line; if there is no foot point on either L_1 or L_2, then P_{a+j} is regarded as P_c.

With P_c found, the distance from P_d to P_c is calculated, which is the estimated tool tip position contour error ε_p.

B. Tool Orientation Contour Error Estimation

For a five-axis CNC machine tool, the machining contour of the workpiece is determined not only by the tool tip position but also by the tool orientation. Therefore, it is important to accurately estimate the orientation contour error, as explained in this section. Particularly, the tool orientation contour error is defined as the angle deviation between the tool axis vector at the tool tip foot point P_c and the desired tool axis vector in this paper. First, the position contour error estimation method in the previous subsection is used to find the tool axis vector O_c at the corresponding tool tip foot point P_c. Then, with the given desired tool axis vector O_d, we can calculate the tool orientation contour error ε_o if the coordinates of the vector O_c are known, as shown in Fig. 5.

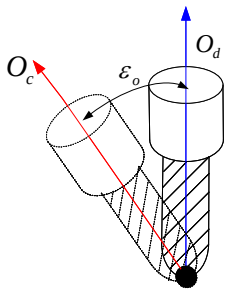


Fig. 5: Estimation of tool orientation contour error

Mathematically, the tool axis vector O_c of the tool tip foot point P_c is calculated as follows:

$$\begin{cases} \vec{O}_{temp} = \vec{O}_{a+j-1} + \frac{\left|\overrightarrow{P_{a+j-1}P_c}\right|}{\left|\overrightarrow{P_{a+j-1}P_{a+j}}\right|}(\vec{O}_{a+j-1} + \vec{O}_{a+j}) \\ \vec{O}_c = \frac{\vec{O}_{temp}}{\left|\vec{O}_{temp}\right|} \end{cases} \tag{3}$$

where O_{a+j-1}, O_{a+j} are the tool axis vector coordinates corresponding to P_{a+i-1}, P_{a+i} on the actual path of the machine tool. Finally, the tool orientation contour error can be calculated as follows:

$$\varepsilon_o = arccos \frac{\vec{O_c} \cdot \vec{O_d}}{\left|\vec{O_c}\right| \left|\vec{O_d}\right|} \tag{4}$$

III. Spatial Iterative Learning Control

After obtaining two estimated contour errors, we design a contour error controller in this section. First, we present the overall control system. Then, we analyze its kinematics and dynamics models, based on which we develop a contour error compensation method by modifying the reference path using sILC.

A. Overall Control System

The control framework of the five-axis CNC machine tool is shown in Fig. 6, which includes four main modules. The first module reads the G code and generates the corresponding output-axis motion; the second module processes the collected data and uses the kinematics model and contour error estimation model to obtain the tool tip contour error and the tool orientation contour error; the third module reduces the contour error by updating the reference path; and the last module evaluates the performance of the CNC system in terms of contour error and reduces it iteratively until the requirement is met.

In the following, we will mainly focus on the kinematics and dynamics modeling, and based on that we will introduce the proposed sILC. It is noteworthy that the contour error is a concept in the spatial domain, so different from the ILC design in the time domain [30], our contour error controller will be developed in the spatial domain. As a result, the geometric reference path is modified to reduce the contour error.

B. Kinematics Modelling Analysis

Consider the kinematics model of a five-axis CNC machine tool as follows:

$$x(t) = \psi(q) \tag{5}$$

where $x(t) \in \mathbb{R}^6$ is the position/orientation coordinate of actual position point P_a in Cartesian space and $q \in \mathbb{R}^5$ is the coordinate in axis space. By differentiating (5) with reference to time, we can get/

$$\dot{x}(t) = J(q)\dot{q} \tag{6}$$

where $\dot{x}(t)$ is the velocity of actual position point P_a in Cartesian space and $J(q) \in \mathbb{R}^{6\times5}$ is the Jacobian matrix. By further differentiating (6), we can get

$$\ddot{x}(t) = \dot{J}(q)\dot{q} + J(q)\ddot{q} \tag{7}$$

$\ddot{x}$ is the acceleration of actual position point P_a in Cartesian space, $\dot{q}$ is the velocity and $\ddot{q}$ is the acceleration in axis space.

Remark 1: Compared with [22], $x(t) \in \mathbb{R}^6$ includes not only the position but also the orientation. Moreover, the method in [22] only analyzes the dynamics model, while the method in this paper needs to combine kinematics and dynamics models for analysis.

C. Dynamics Modelling Analysis

The dynamics model of a CNC system in axis space can be described as follows:

$$M_q\ddot{q}(t) + B_q\dot{q}(t) + F_q = u + d \tag{8}$$

where u is the control input and d is the unknown disturbance applied by the environment. M_q, B_q and F_q are the inertia, coefficients of viscous friction and coulomb friction, respectively.

(6) and (7) can be substituted into the dynamic model (8), so that we have the dynamic model of the five-axis CNC machine tool in Cartesian space, as shown below:

$$M(q)\ddot{x} + B(q,\dot{q})\dot{x} + F(q) = u + d \tag{9}$$

where

$$\begin{aligned} &M(q) = M_q(q)J^+(q), \\ &B(q,\dot{q}) = (B_q(q,\dot{q}) - M_q(q)J^+(q)\dot{J}(q))J^+(q) \\ &F(q) = F_q(q) \end{aligned} \tag{10}$$

with $J^+(q)$ the pseudo inverse of $J(q)$.

Remark 2: Different from the case with a two or three axes as in [22], we need to introduce the pseudo inverse to establish the relationship between the axis space and Cartesian space.

According to [33], there are two properties:

Property 1: Matrix $M(q)$ is symmetric and positive definite.

Property 2: Matrix $2B(q,\dot{q}) - \dot{M}(q)$ is a skew-symmetric matrix, which satisfies $\xi^T(2B(q,\dot{q}) - \dot{M}(q))\xi = 0, \forall\xi \in \mathbb{R}^6$.

Assumption 1: It is assumed that the unknown disturbance from the environment is $d = -\theta$, satisfying $\theta(s) = \theta(s - S)$, where s represents the displacement increment in the repetitive motion and S is the total length of a desired path in one iteration.

The designed controller consists of two parts, namely $u = u_1 + u_2$, where we use u_1 to compensate for the dynamics of the system as below:

$$u_1 = M\ddot{x}_e(t) + B\dot{x}_e(t) + F \tag{11}$$

where $\dot{x}_e$ is defined as an auxiliary variable, as follows:

$$\dot{x}_e = \dot{x}_d - L\varepsilon,\ \varepsilon = x - x_d \tag{12}$$

where x_d represents the coordinate of the desired position point P_d, ε is the position error and L is defined as a positive definite matrix. The other part of the controller u_2 is defined as follows:

$$u_2 = -K\left[\dot{\varepsilon} + L(x - x_r)\right] = -K[e_v + L(x_d - x_r)] \tag{13}$$

where $e_v = \dot{\varepsilon} + L\varepsilon$, x_r represents the coordinate of the reference position point P_r, and K is the position control loop gain.

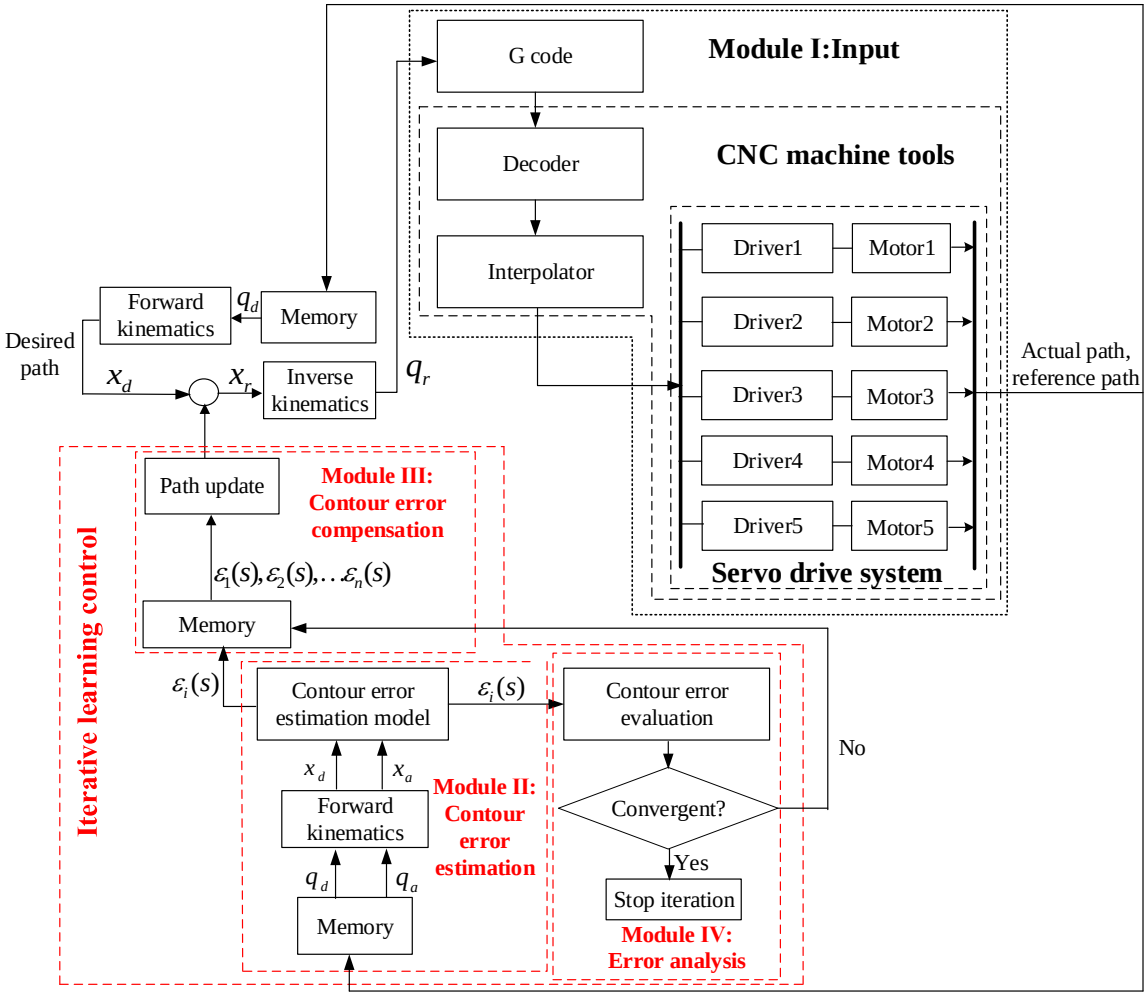


Fig. 6: Control framework, where x represents the coordinate in Cartesian space including position and orientation, q the coordinate in the axis space and ε the contour error, with subscripts r standing for reference, a for actual, d for desired and i for the number of iterations.

By combining (9), (11) and (13), we have

$$M\dot{e}_v(t) + (B+K)e_v(t) = -KL(x_d - x_r) + d \tag{14}$$

From (14), it can be found that if the disturbance d is 0, when the designed reference position x_r is equal to the desired one x_d, then e_v will converge to 0, further leading to $\varepsilon = 0$. However, in practice, the disturbance d is generally not 0, so the reference position x_r needs to be adjusted so that the right half of (14) becomes 0 to achieve $\varepsilon = 0$.

D. Time Domain To Spatial Domain Conversion

As we know, the contour error is a concept in space, but the dynamics model in (14) is in the time domain, so it needs to be converted to the spatial domain. Since time is previously defined as t and the movement increment is s, we link them using the speed along the desired path $v = \frac{ds}{dt}$. Therefore, we can get

$$\frac{d}{dt} = \frac{d}{ds}\frac{ds}{dt} = \frac{d}{ds}v \tag{15}$$

To facilitate the following analysis, we define spatial differentiation as

$$\bigtriangledown = \frac{d}{ds} \tag{16}$$

Then, by converting the model of error dynamics in (14) to the spatial domain, we can get a spatial error dynamics model as follows:

$$Mv \bigtriangledown e_v(s) + (B+K)e_v(s) = -KL[x_d(s) - x_r(s)] + d \tag{17}$$

where $e_v(s)$ is the error in the spatial domain including the defined contour error $\varepsilon(s) = x(s) - x_d(s)$. Therefore, minimization of $e_v(s)$ can lead to minimization of the contour error, which can be achieved through the reference modification method described in the following subsection.

E. Reference Modification Method

According to (17), the contour error $\varepsilon(s)$ can converge to 0 when $KL(x_d - x_r) = d$. Therefore, in the i-th iteration x_r can be designed as below:

$$x_{r,i}(s) = x_d(s) + \frac{1}{KL}\hat{\theta}_i \tag{18}$$

which is equivalent to

$$x_{r,i}(s) = x_{r,i-1}(s) + \frac{1}{KL}\delta\hat{\theta}_i \tag{19}$$

where $\hat{\theta}_i$ represents the estimate of θ_i and will be updated iteratively as below:

$$\delta\hat{\theta}_i = \hat{\theta}_i - \hat{\theta}_{i-1} = \alpha e_{v,i}(s) \tag{20}$$

where α is a positive scalar and represents the learning rate.

According to (17) and (18), in the i-th iteration we can get the error model as below:

$$Mv \bigtriangledown e_{v,i} + (B+K)e_{v,i} = \tilde{\theta}_i \tag{21}$$

where $\tilde{\theta}_i = \hat{\theta}_i - \theta_i$. Furthermore, we can get

$$M \bigtriangledown e_{v,i} + \frac{1}{v}Be_{v,i} + \frac{1}{v}Ke_{v,i} = \frac{1}{v}\tilde{\theta}_i \tag{22}$$

From (22), it is easy to find that if $\tilde{\theta}_i = 0$ then $e_{v,i} \to 0$ and thus $\varepsilon_i \to 0$ which means that the contour error can converge to 0. Therefore, updating $\hat{\theta}_i$ aims at minimizing $\tilde{\theta}_i$, which can be achieved by minimizing the cost function as below:

$$J_c(s) = \frac{1}{2}\int_{s-S}^{s} \frac{1}{\alpha v}\tilde{\theta}_i^T(\tau)\tilde{\theta}_i(\tau)d\tau \tag{23}$$

Accordingly, when the following cost function $J_e(s)$ is minimized, the contour error is minimized:

$$J_e(s) = \frac{1}{2}e_{v,i}^{\mathrm{T}} M e_{v,i} \tag{24}$$

Therefore, we consider to combine $J_c(s)$ and $J_e(s)$. Then we have

$$J = J_c + J_e \tag{25}$$

that will decrease as the number of iterations increases.

F. Convergence analysis

Considering the spatial derivative of J_e, we will get

$$\begin{aligned} &\nabla J_e(s) = e_{v,i}^T M\nabla e_{v,i} + \tfrac{1}{2}e_{v,i}^T \tfrac{1}{v}\dot{M}e_{v,i} \\ &= e_{v,i}^T M\nabla e_{v,i} + e_{v,i}^T \tfrac{1}{v}Be_{v,i} \end{aligned} \tag{26}$$

where we use the skew-symmetry property, i.e. $e_{v,i}^T\dot{M}e_{v,i} = 2e_{v,i}^T Be_{v,i}$. Considering (20), the above equation can be rewritten as

$$\begin{aligned} &\nabla J_e(s) = e_{v,i}^T(M\nabla e_{v,i} + \tfrac{1}{v}Be_{v,i}) \\ &= e_{v,i}^T(\tfrac{1}{v}\tilde{\theta}_i - \tfrac{1}{v}Ke_{v,i}) \end{aligned} \tag{27}$$

Then, the difference between J_c of two consecutive iterations should be considered as below:

$$\begin{aligned} &\Delta J_c = J_c(s) - J_c(s-S) \\ &= \tfrac{1}{2}\textstyle\int_{s-S}^{s}[\tfrac{1}{\alpha v}\tilde{\theta}_i^T\tilde{\theta}_i - \tfrac{1}{\alpha v}\tilde{\theta}_i^T(\tau-S)\tilde{\theta}_i(\tau-S)]d\tau \end{aligned} \tag{28}$$

where we calculate

$$\begin{aligned} &\tfrac{1}{\alpha v}\tilde{\theta}_i^T(\tau)\tilde{\theta}_i(\tau) - \tfrac{1}{\alpha v}\tilde{\theta}_{i-1}^T(\tau-S)\tilde{\theta}_{i-1}(\tau-S) \\ &= (\tfrac{1}{\alpha v}\tilde{\theta}_i^T(\tau)\tilde{\theta}_i(\tau) - \tfrac{1}{\alpha v}\tilde{\theta}_i^T(\tau)\tilde{\theta}_i(\tau-S)) + \\ &(\tfrac{1}{\alpha v}\tilde{\theta}_i^T(\tau)\tilde{\theta}_i(\tau-S) - \tfrac{1}{\alpha v}\tilde{\theta}_{i-1}^T(\tau-S)\tilde{\theta}_{i-1}(\tau-S)) \\ &= -\tfrac{1}{\alpha v}\tilde{\theta}_i^T(\tau)\hat{\theta}_i - \tfrac{1}{\alpha v}\tilde{\theta}_{i-1}^T(\tau-S)\hat{\theta}_i \\ &= -(\tfrac{2}{v}\tilde{\theta}_i^T(\tau) + \delta\hat{\theta}_i)e_{v,i}(\tau) \\ &\le -\tfrac{2}{v}\tilde{\theta}_i^T(\tau)e_{v,i}(\tau) \\ &= -\tfrac{2}{v}e_{v,i}^T\tilde{\theta}_i(\tau) \end{aligned} \tag{29}$$

Thus, we get

$$\Delta J_c \le -\int_{s-S}^{s}[\frac{1}{v}e_{v,i}^T\tilde{\theta}_i(\tau)]d\tau \tag{30}$$

From (27), the variation of J_e can be obtained as follows:

$$\Delta J_e = \int_{s-S}^{s} e_{v,i}^T[\frac{1}{v}\tilde{\theta}_i(\tau) - \frac{1}{v}Ke_{v,i}(\tau)]d\tau \tag{31}$$

By considering (30) and (31), we have

$$\Delta J = \Delta J_c + \Delta J_e \le -\int_{s-S}^{s}\frac{1}{v}e_{v,i}^T(\tau)Ke_{v,i}(\tau)d\tau \tag{32}$$

Because K is a positive scalar, we will get $\Delta J \le 0$.

By (32), it has shown that, when increasing the number of iterations for $s \in [0, S]$, the function J does not increase. Thus, if we can show that J is bounded in the first iteration, we will prove the boundedness of J, i.e., $J_{i=1} < \infty$.

We can consider the spatial derivative of $J_{i=1}$ as below

$$\bigtriangledown J_{i=1} = \bigtriangledown J_c + \bigtriangledown J_e \tag{33}$$

According to (32), we have

$$\bigtriangledown J_{i=1} \le -\frac{1}{v}e_{v,i}^T Ke_{v,i} \le 0 \tag{34}$$

Therefore, we integrate $\bigtriangledown J_{i=1}$ from 0 to s and will obtain

$$J_{i=1} - J_{i=1}(0) \le 0 \tag{35}$$

It is known that $e_{v,1}(0)$ and M are bounded, so according to (24) we can deduce that $J_e(0)$ is also bounded. Since $\hat{\theta}$ is initialized as 0 before the first iteration and true values of parameters θ and the displacement S are bounded, $J_c(0)$ is bounded. Thus, we can get that $J_{i=1}(0)$ is bounded, and so $J_{i=1}$ can be inferred to be bounded.

Finally, according to (32), we get

$$\Delta J \le -\int_{s-S}^{s}\frac{1}{v}e_{v,i}^T Ke_{v,i}d\tau \tag{36}$$

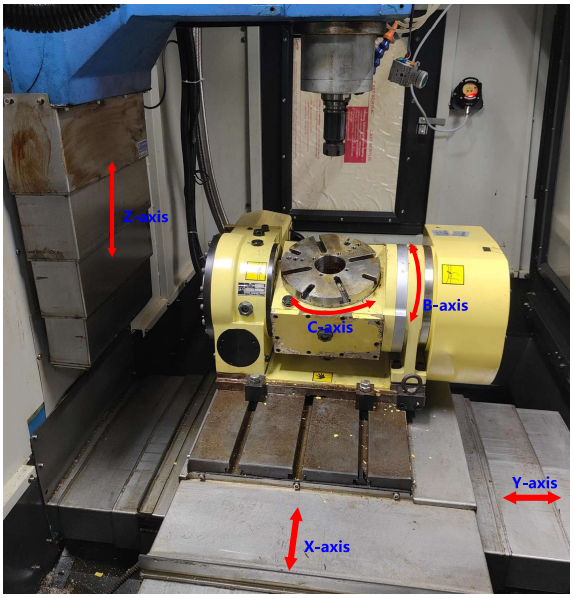
From the above inequality, we can obtain

$$J - J_{i=1} \le -\sum_{j=1}^{i-1}\int_{s-S}^{s}\frac{1}{v}e_{v,i}^T Ke_{v,i}d\tau \tag{37}$$

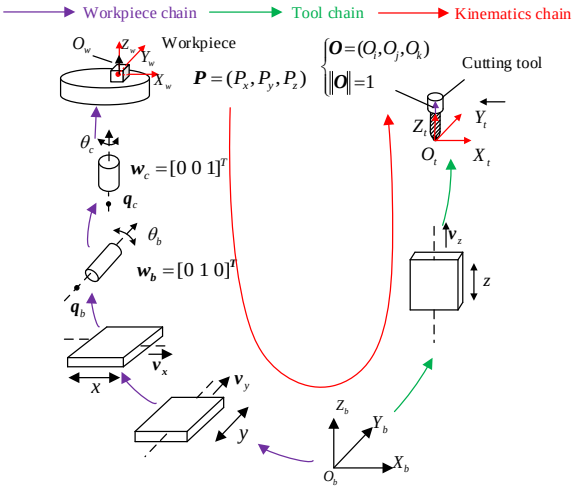
which will lead to

$$J_{i=1} \ge \sum_{j=1}^{i-1}\int_{s-S}^{s}\frac{1}{v}e_{v,i}^T Ke_{v,i}d\tau \tag{38}$$

Because $J_{i=1}$ is bounded, we can get the conclusion that $\|e_{v,i}\| \to 0$ when the iteration number $i \to \infty$.



(a)



(b)

Fig. 7: The experimental machine: (a) the five-axis CNC machine; (b) kinematics chains (adapted from [34]).

IV. Experiments

A. Experimental Setup

The effectiveness and reliability of the five-axis contour error control algorithm are verified on the experimental platform shown in Fig. 7, which is a BC type five-axis CNC machine tool. The axes X, Y and Z are translational and the axes B, C are rotational. Axis B rotates about axis Y and axis C rotates about axis Z. The platform uses a Googol controller, the model of the three linear axis drives is Yaskawa SGDM-10ADA-V, and the model of the rotary axis drives is Sanyo RS1A03AAWA. The position loop gains of the three linear axes are respectively set to $K_{vx} = 150, K_{vy} = 199, K_{vz} = 107$, and the position loop gains of the two rotary axes are respectively set to $K_{vb} = 75, K_{vc} = 50$. The pitch of the motor is $5mm$ and the resolution of the encoder of the rotating shaft is $32768p/r$. On the control panel, the sampling frequency is $500Hz$. The input of CNC machine tool is G code with G01, and it collects the reference and actual position data of each axis.

We use the following two error indicators to evaluate the effectiveness of the five-axis contour error control strategy:

1) $\varepsilon_{\max} = \max|\varepsilon(s)|$, the maximum (MAX) absolute value of the two contour errors over the whole path for transient contouring performance evaluation.
2) $\varepsilon_{rms} = \sqrt{\frac{\sum_{n=1}^{N}\varepsilon^2(n)}{N}}$, the root mean square (RMS) value of the two contour errors $\varepsilon(s)$ for steady-state performance evaluation.

We use (20), (21) and (22) to update the reference path with a fixed learning rate α.

B. Experimental Results

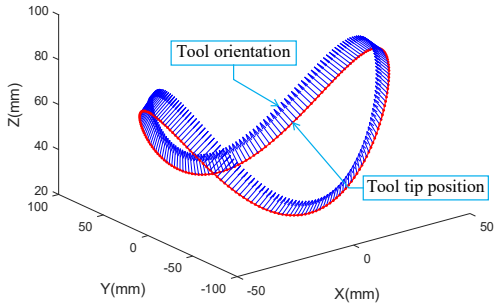


Fig. 8: Three-dimensional saddle surface path

1) Contour Error Compensation on Saddle Surface Path: The saddle surface path shown in Fig. 8 is used in the first experiment. First, we use tracking error control to reduce contour error. When the default feed rate is $1000mm/min$ and $\alpha = 0.5$, the compensated reference path is converted into G code through the inverse kinematics and sent to the CNC machine tool. The contour error estimation result is shown in Fig. 9, showing that the tracking error control can effectively reduce the contour error.

It can be seen from Tab. I that after multiple iterations of compensation for the contour error, the MAX value of the tool tip position contour error is reduced from 8.656μm to 2.628μm, which is reduced by 69.64%; the RMS value of the tool tip position contour error is reduced from 5.380μm to 0.830μm, which is reduced by 84.57%; the MAX value of the tool orientation contour error is reduced from $3.29\times10^{-4}rad$ to $3\times10^{-5}rad$, which is reduced by 90.88%; and the RMS value of tool orientation contour error is reduced from $2.993\times10^{-4}rad$ to $7.043\times10^{-6}rad$, which is reduced by 97.65%.

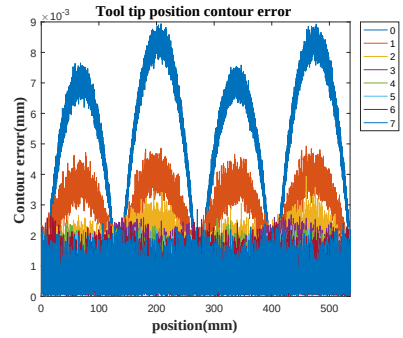
Then, we implement the contour error control based on sILC proposed in this paper. With the default feed rate $1000mm/min$ and the iterative learning rate $\alpha = 0.5$, the experimental result is shown in Fig. 10. From Fig. 10(a), it is shown that after the first iteration, the tool tip position contour error and the tool orientation contour error have been significantly reduced, and after 7 iterations, the tool

TABLE I: Contour error comparison with two control methods for saddle surface path

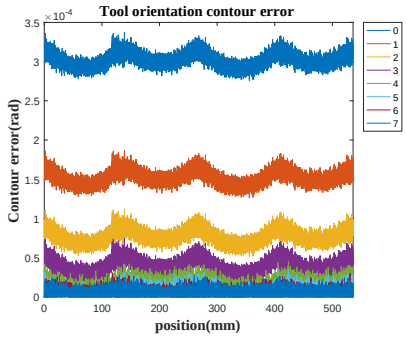
	$\varepsilon_{p,MAX}/\mu$m	$\varepsilon_{p,RMS}/\mu$m	$\varepsilon_{o,MAX}/rad$	$\varepsilon_{o,RMS}/rad$
Initial control error	8.656	5.380	3.29×10^{-4}	2.993×10^{-4}
Final control error with ILC tracking error control	2.628	0.830	3×10^{-5}	7.043×10^{-6}
Final control error with sILC contour error control	2.399	0.755	2.512×10^{-5}	6.845×10^{-6}

tip position contour error and the tool orientation are reduced to a converging value.

Furthermore, after 7 iterations of compensation, the MAX value of the tool tip position contour error is reduced from 8.656μm to 2.399μm, which is reduced by 72.29%; the RMS value of the tool tip position contour error is reduced from 5.380μm to 0.755μm, which is reduced by 85.97%; the MAX value of the tool orientation contour error is reduced from $3.29 \times 10^{-4} rad$ to $2.512 \times 10^{-5} rad$, which is reduced by 92.36%; and the RMS value of tool orientation contour error is reduced from $2.993 \times 10^{-4} rad$ to $6.845 \times 10^{-6} rad$, which is reduced by 97.71%.



(a)

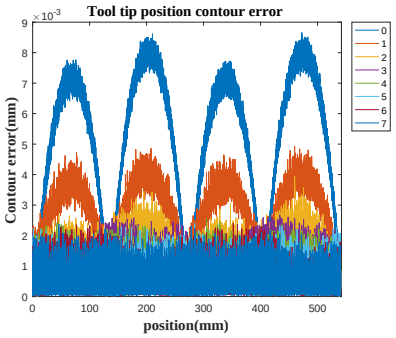


(b)

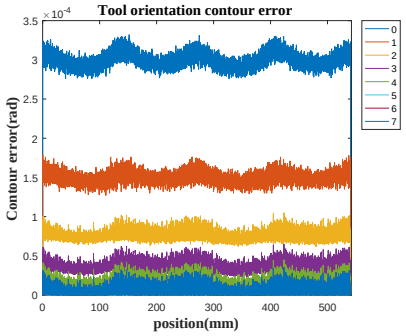
Fig. 9: Tool tip position contour error (a) and tool orientation contour error (b) under tracking error control. $0, \ldots, 7$ stand for the iteration number.

Compared with tracking error control, the results show the sILC control method performs better in contour error control, as further illustrated in Fig. 11.

2) Contour Error Compensation on Butterfly Path: Since the saddle surface path is relatively simple, the second experiment is designed with a more complex butterfly path. This path is based on the NURBS curve and projected onto the saddle surface to generate a five-axis butterfly tool path as shown in Fig. 12. In this case, we



(a)



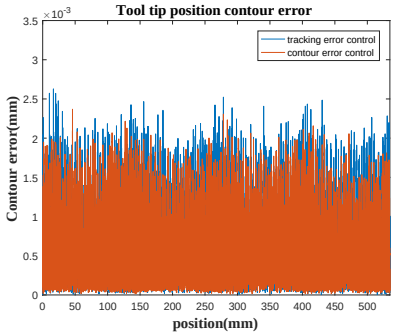
(b)

Fig. 10: Tool tip position contour error (a) and tool orientation contour error (b) under contour error control. $0, \ldots, 7$ stand for the iteration number.

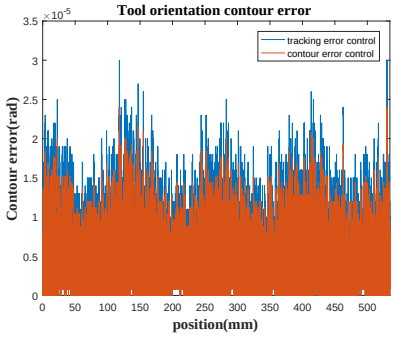
set the default feed rate to $1500 mm/min$ and the iterative learning rate $\alpha = 0.5$.

The results of tracking error control are presented in Fig. 13. Moreover, Tab. II shows that after multiple iterations of compensation for the contour error, the MAX value of the tool tip position contour error is reduced from 49.633μm to 36.992μm, which is reduced by 25.47%; the RMS value of the tool tip position contour error is reduced from 26.664μm to 2.591μm, which is reduced by 90.28%; the MAX value of the tool orientation contour error is reduced from $7.45 \times 10^{-4} rad$ to $4.45 \times 10^{-4} rad$, which is reduced by 40.27%; and the RMS value of tool orientation contour error is reduced from $3.139 \times 10^{-4} rad$ to $3.782 \times 10^{-5} rad$, which is reduced by 87.95%.

Then, the proposed contour error control method is verified, whose performance is illustrated in Fig. 14. Moreover, Tab. II shows that after multiple iterations of compensation for the contour error, the MAX value of the tool tip position contour error is reduced from from 49.633μm to 15.376μm, which is reduced by 69.02%; the RMS value of the tool tip position contour error is

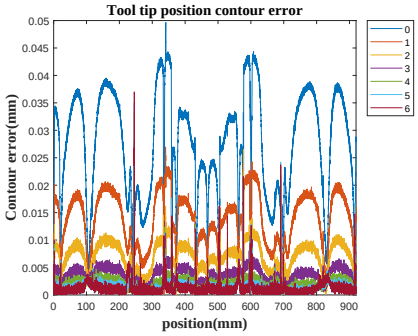


(a)

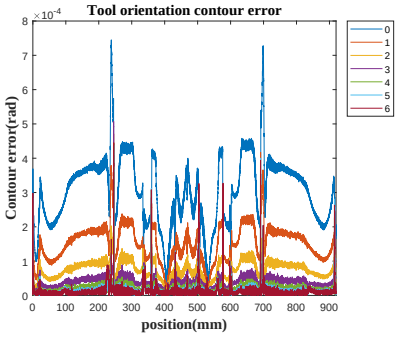


(b)

Fig. 11: Tool tip position contour error (a) and tool orientation contour error (b) after compensation by two control methods.



(a)



(b)

Fig. 13: Tool tip position contour error (a) and tool orientation contour error (b) of a butterfly path under tracking error control. $0, \ldots, 6$ stand for the iteration number.

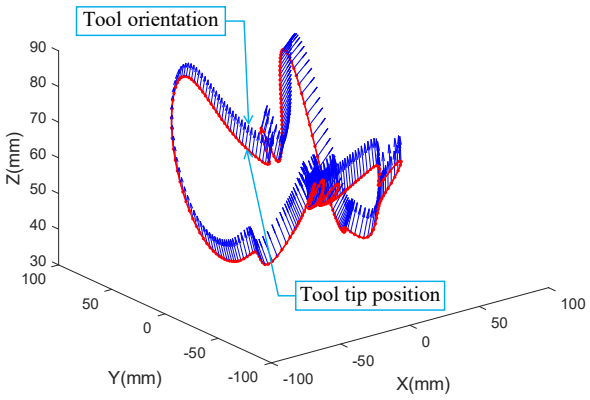


Fig. 12: Three-dimensional butterfly path

reduced from 26.664μm to 0.896μm, which is reduced by 96.64%; the MAX value of the tool orientation contour error is reduced from $7.45 \times 10^{-4} rad$ to $2.59 \times 10^{-4} rad$, which is reduced by 65.23%; and the RMS value of tool orientation contour error is reduced from $3.139 \times 10^{-4} rad$ to $1.285 \times 10^{-5} rad$, which is reduced by 95.91%.

Furthermore, we compare the final performance of the two control methods, as illustrated in Fig. 15. From the comparison results, it can be seen that the control method proposed in this paper has better performance in control of both tool tip position contour error and tool orientation contour error.

3) Machining Efficiency: It is usually argued that high-speed and high-precision are two conflicting objectives, i.e. high-speed is achieved at the cost of low-precision and vice versa. In this section, we present preliminary experimental results to demonstrate that the proposed learning control strategy may resolve this issue and can realize high-speed, high-precision control.

With other settings kept the same as in the above experiments, we increase the feed rate to $3000 mm/min$ (F3000) and the iterative learning rate α to 0.5. The tool tip position contour error and tool orientation contour error are compensated separately. After 6 iterations, we can get the final result as shown in Fig. 16.

The contour errors with the feed rate $1500 mm/min$ without compensation (F1500) is also presented for comparison. In order to compare the performance in two conditions more clearly, the MAX and RMS values of the contour errors are summarized in Tab. III.

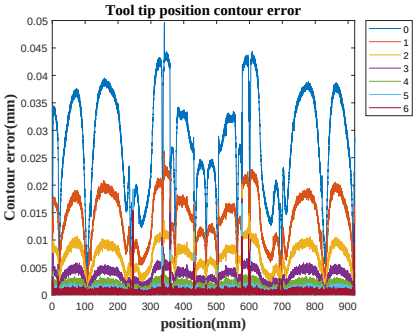
From Fig. 16 and Tab. III, it can be seen that after 6 iterations of learning control, the MAX value of tool tip position contour error is reduced by 26.98% compared with the MAX value tool tip position contour error with F1500 but without compensation. The RMS value of tool tip position contour error is reduced by 88.19%, the MAX value of tool orientation contour error is reduced by 18.06%, and the RMS value of tool orientation contour error is reduced by 84.94%. It is important to note that

TABLE II: Contour error comparison with two control methods for butterfly path

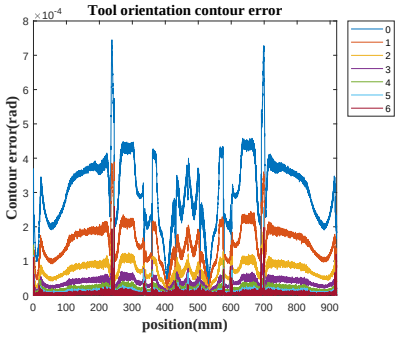
	$\varepsilon_{p,MAX}/\mu$m	$\varepsilon_{p,RMS}/\mu$m	$\varepsilon_{o,MAX}/rad$	$\varepsilon_{o,RMS}/rad$
Initial error	49.633	26.664	7.45×10^{-4}	3.139×10^{-4}
Error with ILC tracking error control	36.992	2.591	4.45×10^{-4}	3.782×10^{-5}
Error with sILC contour error control	15.376	0.896	2.59×10^{-4}	1.285×10^{-5}

TABLE III: Contour error comparison with F1500 and F3000

	$\varepsilon_{p,MAX}/\mu$m	$\varepsilon_{p,RMS}/\mu$m	$\varepsilon_{o,MAX}/rad$	$\varepsilon_{o,RMS}/rad$
F1500	49.633	26.664	7.45×10^{-4}	3.139×10^{-4}
F3000	36.244	3.149	6.104×10^{-4}	4.727×10^{-5}

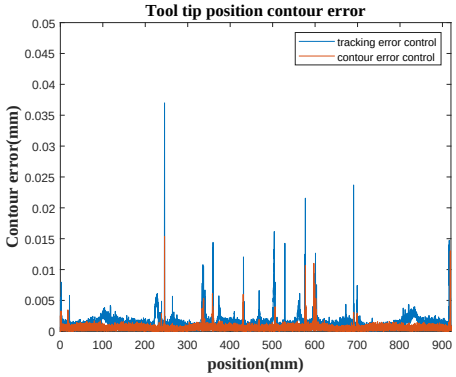


(a)

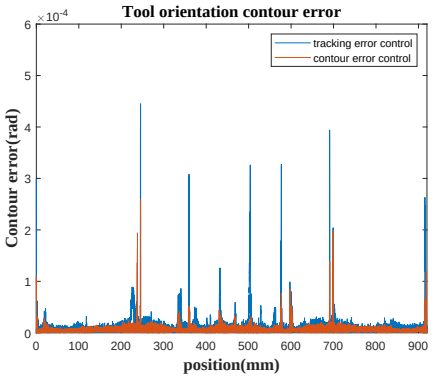


(b)

Fig. 14: Tool tip position contour error (a) and tool orientation contour error (b) of a butterfly path under contour error control. $0, \ldots, 6$ stand for the iteration number.



(a)



(b)

Fig. 15: Tool tip position contour error (a) and tool orientation contour error (b) of a butterfly path after compensation by two control methods.

these reductions of contour errors are achieved with the machining efficiency doubled. Therefore, the proposed control strategy provides a solution for high-speed, high-precision machining, instead of finding a trade-off between efficiency and machining accuracy.

V. Conclusion

In this work, we analyze the characteristics of five-axis machine tools, and propose a five-axis contour error compensation strategy based on spatial iterative learning control (sILC). Then we consider the kinematics and dynamics of the five-axis machine tool, and theoretically prove the stability and convergence of the proposed compensation strategy. Finally, we verify the proposed compensation strategy by using a BC type five-axis CNC machine tool and compare the control result with the tracking error iterative learning control. The experimental results show that the contour error control method based on sILC can significantly reduce the five-axis tool tip position contour error and tool orientation contour error, while it maintains the machining efficiency. The method proposed in this article is an off-line control method, which can only be applied to repetitive machining tasks. For a new trajectory, the proposed method needs a new sILC process to achieve the control effect, making it time-consuming. Nerual network is widely utilized for dynamics modeling and has been successfully applied in various

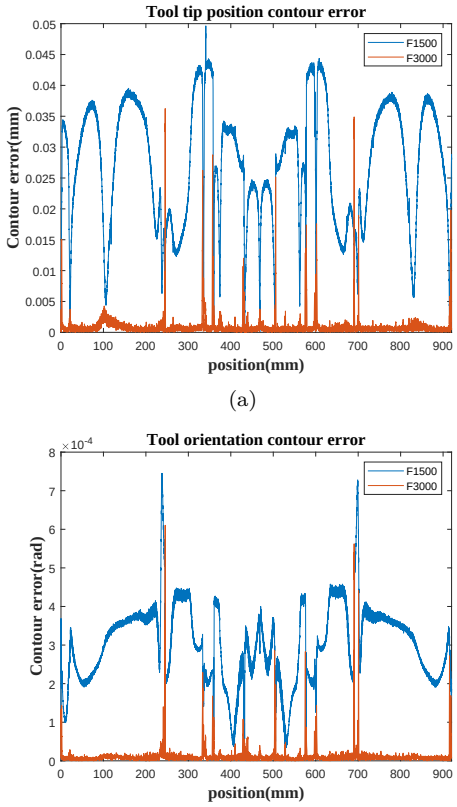


Fig. 16: Tool tip position contour error (a) and tool orientation contour error (b) with F1500 and F3000.

fields [35], [36], [37], [38]. In the future work, we will use neural network to establish the model of the machine tool, and then the proposed method will be applied to this model rather than the physical machine tool. This will greatly reduce the processing time if the accuracy of the model is high, and lead to online learning that is preferable for non-repetitive tasks.

References

[1] L. Tang and R. G. Landers, "Multiaxis contour control—the state of the art," IEEE Transactions on Control Systems Technology, vol. 21, no. 6, pp. 1997–2010, 2013.

[2] R. Ramesh, M. Mannan, and A. Poo, "Tracking and contour error control in cnc servo systems," International Journal of Machine Tools and Manufacture, vol. 45, no. 3, pp. 301–326, 2005.

[3] Z. Y. Jia, J. W. Ma, D. N. Song, F. J. Wang, and W. Liu, "A review of contouring-error reduction method in multi-axis cnc machining," International Journal of Machine Tools and Manufacture, vol. 125, pp. 34–54, 2018.

[4] N. Wang, C. Chen, and A. Di Nuovo, "A framework of hybrid force/motion skills learning for robots," IEEE Transactions on Cognitive and Developmental Systems, vol. 13, no. 1, pp. 162–170, 2020.

[5] Y. Tao, Z. Zhu, Q. Xu, H.-X. Li, and L. Zhu, "Tracking control of nanopositioning stages using parallel resonant controllers for high-speed nonraster sequential scanning," IEEE Transactions on Automation Science and Engineering, 2020.

[6] G. Peng, C. P. Chen, and C. Yang, "Neural networks enhanced optimal admittance control of robot-environment interaction using reinforcement learning," IEEE Transactions on Neural Networks and Learning Systems, 2021.

[7] D. Huang, C. Yang, Y. Pan, and L. Cheng, "Composite learning enhanced neural control for robot manipulator with output error constraints," IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 209–218, 2019.

[8] Y. Altintas, K. Erkorkmaz, and W.-H. Zhu, "Sliding mode controller design for high speed feed drives," CIRP Annals, vol. 49, no. 1, pp. 265–270, 2000.

[9] K. Erkorkmaz and Y. Altintas, "High speed cnc system design. part iii: high speed tracking and contouring control of feed drives," International Journal of Machine Tools and Manufacture, vol. 41, no. 11, pp. 1637–1658, 2001.

[10] M. Yang, J. Yang, and H. Ding, "A two-stage friction model and its application in tracking error pre-compensation of cnc machine tools," Precision Engineering, vol. 51, pp. 426–436, 2018.

[11] J. Yang and Z. Li, "A novel contour error estimation for position loop-based cross-coupled control," IEEE/ASME transactions on mechatronics, vol. 16, no. 4, pp. 643–655, 2010.

[12] K. Zhang, A. Yuen, and Y. Altintas, "Pre-compensation of contour errors in five-axis cnc machine tools," International Journal of Machine Tools and Manufacture, vol. 74, pp. 1–11, 2013.

[13] Y. Koren and C.-C. Lo, "Variable-gain cross-coupling controller for contouring," CIRP annals, vol. 40, no. 1, pp. 371–374, 1991.

[14] H. Zhao, L. Zhu, and H. Ding, "Cross-coupled controller design for triaxial motion systems based on second-order contour error estimation," Science China Technological Sciences, vol. 58, no. 7, pp. 1209–1217, 2015.

[15] Y.-T. Shih, C.-S. Chen, and A.-C. Lee, "A novel cross-coupling control design for bi-axis motion," International Journal of machine tools and manufacture, vol. 42, no. 14, pp. 1539–1548, 2002.

[16] Y. Chen, D. Huang, Y. Li, and X. Feng, "A novel iterative learning approach for tracking control of high-speed trains subject to unknown time-varying delay," IEEE Transactions on Automation Science and Engineering, 2020.

[17] M. Minakais, S. Mishra, and J. T. Wen, "Database-driven iterative learning for building temperature control," IEEE Transactions on Automation Science and Engineering, vol. 16, no. 4, pp. 1896–1906, 2019.

[18] B. Wally, J. Vyskočil, P. Novák, C. Huemer, R. Šindelář, P. Kadera, A. Mazak-Huemer, and M. Wimmer, "Leveraging iterative plan refinement for reactive smart manufacturing systems," IEEE Transactions on Automation Science and Engineering, vol. 18, no. 1, pp. 230–243, 2020.

[19] Z. Lu, N. Wang, and C. Yang, "A novel iterative identification based on the optimised topology for common state monitoring in wireless sensor networks," International Journal of Systems Science, pp. 1–15, 2021.

[20] K. L. Barton and A. G. Alleyne, "A cross-coupled iterative learning control design for precision motion control," IEEE Transactions on Control Systems Technology, vol. 16, no. 6, pp. 1218–1231, 2008.

[21] M.-S. Tsai, M.-T. Lin, and H.-T. Yau, "Development of command-based iterative learning control algorithm with consideration of friction, disturbance, and noise effects," IEEE Transactions on Control Systems Technology, vol. 14, no. 3, pp. 511–518, 2006.

[22] J. Li, Y. Wang, Y. Li, and W. Luo, "Reference trajectory modification based on spatial iterative learning for contour control of two-axis nc systems," IEEE/ASME Transactions on Mechatronics, vol. 25, no. 3, pp. 1266–1275, 2020.

[23] Y. Zhang and K. Tang, "Automatic sweep scan path planning for five-axis free-form surface inspection based on hybrid swept area potential field," IEEE Transactions on Automation Science and Engineering, vol. 16, no. 1, pp. 261–277, 2018.

[24] Y. Zhang, L. Zhu, P. Zhao, P. Hu, and X. Zhao, "Skeleton curve-guided five-axis sweep scanning for surface with multiple holes," IEEE Transactions on Automation Science and Engineering, 2021.

[25] X. Li, H. Zhao, X. Zhao, and H. Ding, "Dual sliding mode contouring control with high accuracy contour error estimation for five-axis cnc machine tools," International Journal of Machine Tools and Manufacture, vol. 108, pp. 74–82, 2016.

[26] Q. Hu, Y. Chen, and J. Yang, "On-line contour error estimation and control for corner smoothed five-axis tool paths," International Journal of Mechanical Sciences, vol. 171, p. 105377, 2020.

[27] M. Yang, J. Yang, and H. Ding, "A high accuracy on-line estimation algorithm of five-axis contouring errors based on three-point arc approximation," International Journal of Machine Tools and Manufacture, vol. 130, pp. 73–84, 2018.
[28] M. R. Khoshdarregi, S. Tappe, and Y. Altintas, "Integrated five-axis trajectory shaping and contour error compensation for high-speed cnc machine tools," IEEE/ASME Transactions on Mechatronics, vol. 19, no. 6, pp. 1859–1871, 2014.
[29] S. Yang, A. H. Ghasemi, X. Lu, and C. E. Okwudire, "Pre-compensation of servo contour errors using a model predictive control framework," International Journal of Machine Tools and Manufacture, vol. 98, pp. 50–60, 2015.
[30] M.-T. Lin, B.-H. Tsai, C.-Y. Lee, S.-K. Wu, and H.-M. Lee, "Command-based iterative learning control with tcp function for five-axis contouring," in 2013 IEEE International Conference on Mechatronics and Automation, pp. 1027–1032, IEEE, 2013.
[31] T.-Q. Duong, P. Rodriguez-Ayerbe, S. Lavernhe, C. Tournier, and D. Dumur, "Contour error pre-compensation for five-axis high speed machining: offline gain adjustment approach," The International Journal of Advanced Manufacturing Technology, vol. 100, no. 9, pp. 3113–3125, 2019.
[32] N. Uchiyama et al., "Estimation of tool orientation contour errors for five-axismachining," Robotics and Computer-Integrated Manufacturing, vol. 29, no. 5, pp. 271–277, 2013.
[33] S. S. Ge, T. H. Lee, and C. J. Harris, Adaptive Neural Network Control of Robotic Manipulators. WORLD SCIENTIFIC, 1998.
[34] J. Yang and Y. Altintas, "A generalized on-line estimation and control of five-axis contouring errors of cnc machine tools," International Journal of Machine Tools and Manufacture, vol. 88, pp. 9–23, 2015.
[35] R. d. S. B. Ferreira, A. Sabbaghi, and Q. Huang, "Automated geometric shape deviation modeling for additive manufacturing systems via bayesian neural networks," IEEE Transactions on Automation Science and Engineering, vol. 17, no. 2, pp. 584–598, 2019.
[36] W. Qi, H. Su, and A. Aliverti, "A smartphone-based adaptive recognition and real-time monitoring system for human activities," IEEE Transactions on Human-Machine Systems, vol. 50, no. 5, pp. 414–423, 2020.
[37] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. De Momi, "Deep neural network approach in robot tool dynamics identification for bilateral teleoperation," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2943–2949, 2020.
[38] B. Xiao, W. Xu, J. Guo, H.-K. Lam, G. Jia, W. Hong, and H. Ren, "Depth estimation of hard inclusions in soft tissue by autonomous robotic palpation using deep recurrent neural network," IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 1791–1799, 2020.

Jiangang Li received the BEng, MEng, PhD degrees from the Xi'an Jiaotong University, China, in 1999, 2002 and 2005, respectively. From 2007 to present, he has been an Associate Professor in control science and engineering with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, China. From 2015 to 2016, he has been a Visiting Associate in computing and mathematical sciences with the California Institute of Technology. His general research interests include high speed and high performance control system design, motion control and motion planning.

Zhiyang You received the BEng degree from the Harbin University of Science and Technology, Harbin, China, in 2019. He is currently working towards the MEng degree in the Harbin Institute of Technology Shenzhen, China. His research interests include motion control and high-performance control system design.

Yanan Li (S'10-M'14-SM'21) received the BEng and MEng degrees from the Harbin Institute of Technology, China, in 2006 and 2008, respectively, and the PhD degree from the National University of Singapore, in 2013. Currently he is a Senior Lecturer in Control Engineering with the Department of Engineering and Design, University of Sussex, UK. From 2015 to 2017, he was a Research Associate with the Department of Bioengineering, Imperial College London, UK. From 2013 to 2015, he was a Research Scientist with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore. His general research interests include human-robot interaction, robot control and control theory and applications.

Enming Miao received the M.S. and Ph.D. degrees from the Hefei University of Technology, in 1999 and 2004, respectively. He is a professor in Mechanical Engineering at Chongqing University of Technology, Chongqing, China. He serves as the special Professor of Chongqing Bayu scholars and the Discipline Leader of Chongqing. He has published two monographs and more than 150 papers, obtained nine inventions and drafted 4 provincial and ministerial local standards. His research interests include precision control technology and robust modeling technology of intelligent manufacturing, functional measurement and control technology of numerical control equipment, remote operation and maintenance of equipment and accuracy theory.

Ruijie Yue received the BEng degree from the Harbin Institute of Technology Weihai, Shandong, China, in 2020. She is currently working towards the MEng degree in the Harbin Institute of Technology Shenzhen, China. Her research interests include motion control and high-performance control system design.